

CPP 程式設計題

命題者：Huang

題目名稱(中文/英文)： 矩形/Rect

主要測試觀念： OPERATOR OVERLOADING

Basics

- ☒ C++ BASICS 1
- ☒ FLOW OF CONTROL
- ☐ FUNCTION BASICS
- ☐ PARAMETERS AND OVERLOADING
- ☐ ARRAYS
- ☒ STRUCTURES AND CLASSES
- ☐ CONSTRUCTORS AND OTHER TOOLS
- ☒ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES
- ☐ STRINGS
- ☐ POINTERS AND DYNAMIC ARRAYS

Functions

- ☐ SEPARATE COMPILATION AND NAMESPACES
- ☐ STREAMS AND FILE I/O
- ☐ RECURSION
- ☐ INHERITANCE
- ☐ POLYMORPHISM AND VIRTUAL FUNCTIONS
- ☐ TEMPLATES
- ☐ LINKED DATA STRUCTURES
- ☐ EXCEPTION HANDLING
- ☐ STANDARD TEMPLATE LIBRARY
- ☐ PATTERNS AND UML

題目說明：

在 2D 橫向卷軸遊戲中，一個角色的大小及位置常以一個矩形表示，利用兩矩形重疊能夠計算兩角色是否有碰撞；找出包含兩矩形的最小矩形，則能作為相機位置的參考。請實作一個 Rect 的 class，包含以下功能：

(請自行創建並加入 Rect.cpp 與 Rect.h 兩檔於繳交的壓縮檔案內，並以 Rect.h 進行 class 宣告；Rect.cpp 進行 class 實作，而不更改附檔的 main.cpp 中之內容，否則將斟酌扣分。)

- 1 以 **private member** 宣告 x 與 y 與 width 與 height，分別為矩形左上角的 x 值與 y 值，以及寬和高。
- 2 輸入矩形的 x, y, width, height，建立矩形實體。
- 3 利用運算子+，計算能包含兩矩形最小矩形，並回傳結果矩形。程式能以 rect1 + rect2 運算。
- 4 利用運算子*，計算兩矩形的重疊範圍(兩矩形有任區域，線段或點重疊則視為重疊)，並回傳結果值，若未重疊則需另外直接輸出” Not overlapping”，並回傳預設值。程式能以 rect1 * rect2 運算。
- 5 實作 toString()功能，回傳字串格式：(x 值， y 值， 寬， 高)。
- 6 預設矩形的參數值為:左上角座標(0,0)，長寬皆為 0 的矩形。

輸入說明：

每組測資分別輸入兩矩形在二維直角座標系中的左上角點的座標以及長寬：

範例輸入中，第一行輸入分別為 rect1 的左上角點的座標的 x 值，y 值，以及 width(寬)和 height(高)；第二行輸入分別為 rect2 的左上角點的座標的 x 值，y 值，以及寬和高。

所有輸入皆為 int。輸入 x 範圍：-100 <= x <= 100；y 範圍：-100 <= y <= 100；0 <= width <= 100；0 <= height <= 100。

輸出說明：

分別計算出包含兩矩形的最小矩形(unionRect)，以及兩矩形的疊合範圍(interRect)。

輸出 unionRect 的左上角點的座標 x 值與 y 值，width 與 height。

輸出 interRect 的左上角點的座標 x 值與 y 值，width 與 height。

I0 範例：

	Sample Input	Sample Output
第一組	0 0 5 5 1 1 10 10	union : (0 , 1 , 11 , 10) intersection : (1 , 0 , 4 , 5)
第二組	0 0 0 0 1 1 1 1	union : (0 , 1 , 2 , 1) Not overlapping intersection : (0 , 0 , 0 , 0)
第三組	-5 5 5 10 0 5 5 10	union : (-5 , 5 , 10 , 10) intersection : (0 , 5 , 0 , 10)

附屬資料：

☒ 解答程式：source.cpp

☒ 測試資料：input.txt, output.txt

☒ 易，僅需用到基礎程式設計語法與結構

☐ 中，需用到多項程式設計語法與結構

☐ 難，需用到多項程式結構或較為複雜之資料型態或結構

解題時間：20 分鐘。

其他註記：

測資共六組，前三組皆對者得 70%，後三組每對一組得 10%。